User's Guide

For Safety information, Warranties, and Regulatory information, see the pages behind the index.

XRAY
MasterWorks™

HP Connection for Microtec Research MasterWorks XRAY HP ICE

HP Connection for Microtec Research MasterWorks XRAY HP ICE—At a Glance

The HP Connection for Microtec Research MasterWorks XRAY HP ICE lets you use XRAY MasterWorks with Hewlett-Packard emulators, XRAY MasterWorks is a product of Microtec Research, Inc.

The XRAY debugger

The connection software adds a new icon to the XRAY MasterWorks control panel: xhi68khp. The xhi68khp tool starts an XRAY debugger which is connected to your target system through an HP 64700-series emulator or an HP E3490A or HP E3458A Processor Probe. Throughout this manual, this debugger is called "the XRAY debugger."

The HP emulator/analyzer graphical user interface

If you have an HP 64700-series emulator, you can use the HP emulator/analyzer graphical interface. This interface is not available for the HP Processor Probe.

Use the emulator/analyzer graphical interface when you need to analyze bus activity, when you need to set sophisticated triggers to stop execution or capture trace data, and when you need to work with emulator copies of internal registers (EMSIM registers).
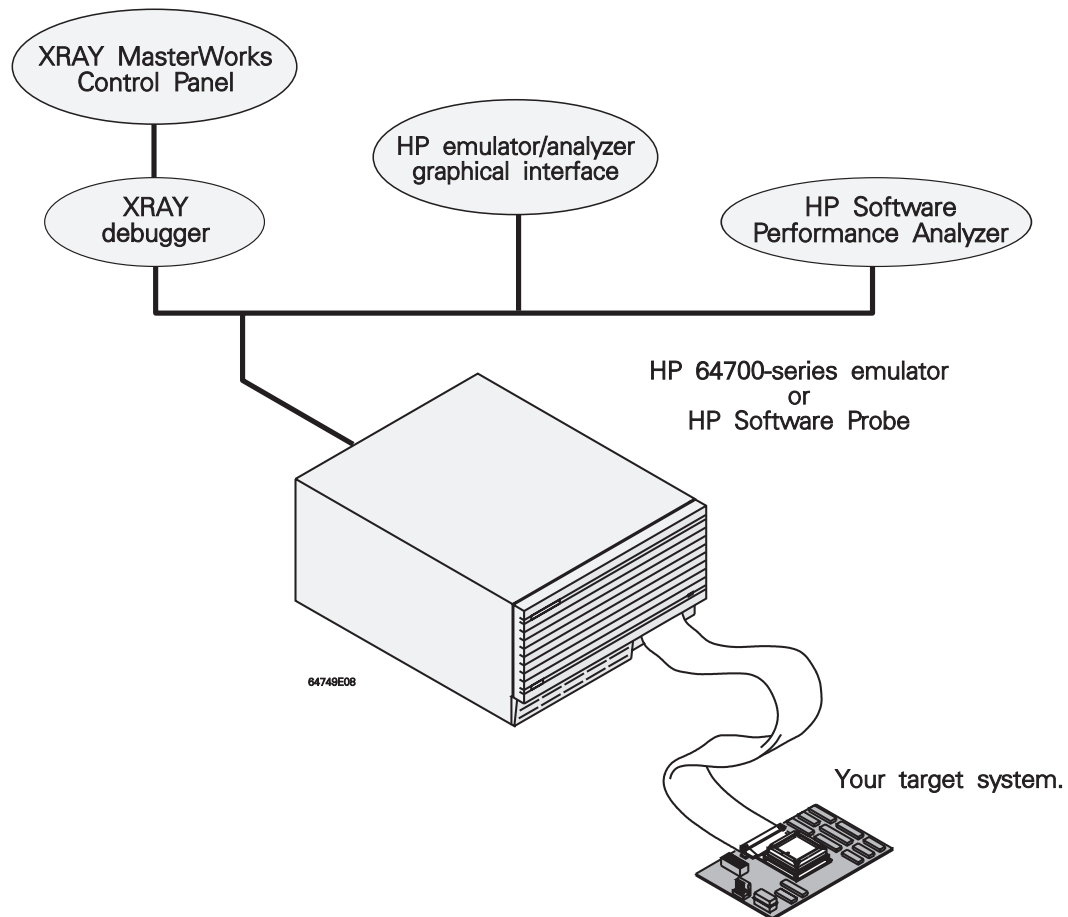
The HP Software Performance Analyzer

If you have an HP 64700-series emulator, you can also use the HP B1487 Software Performance Analyzer to measure program activity and module duration.

The HP Branch Validator

If you have an HP 64700-series emulator, you can also use the HP B1418 Branch Validator to measure the extent to which branches are executed during runs of your target program.

Interfaces working together

You can use the HP emulator/analyzer graphical interface, the HP Software Performance Analyzer, and the HP Branch Validator alongside the XRAY debugger.

XRAY MasterWorks
Control Panel

XRAY
debugger

HP emulator/analyzer
graphical interface

HP Software
Performance Analyzer

HP 64700-series emulator
or
HP Software Probe

64749E08

Your target system.

Contents

Contents

3 Configuration

4 Interactions Between Interfaces

5 Restrictions and Limitations

6 Solving Problems

7 About this Version

Contents

1

Getting Started

Introduction

This manual

This manual should be used as a supplement to the Microtec Research XRAY Debugger manuals. With the exception of the tutorial, this manual does not duplicate information which can be found in the Microtec Research manuals.

You should have the following Microtec Research manuals available for reference:

- *XRAY Debugger Reference Manual*
- *XRAY Debugger Supplement for the 68000 Family*
- *Integrated Environment Utilities*
- *Product Installation (UNIX Hosts)*

HP XRAY ICE

The HP XRAY ICE Debugger is a product developed jointly by Microtec Research and the Hewlett-Packard Company. To have a functional debugger, you need to install software from both companies.

xhi68khp

The xhi68khp portion of the software is supplied by Microtec Research.

xti68khp

The xti68khp portion of the software is supplied by Hewlett-Packard. This process connects the XRAY Debugger user interface to the HP emulation hardware.

Product licensing

A license for XHI68KHP or XOI68KHP must be included in your license file. Refer to your Microtec Research License Authorizaiton form and the *Microtec Research Product Installation (UNIX Hosts)* document for details.

A license for XHS68K is not required to use the XRAY debugger with the HP connection software. A demo license for XHS68K is supplied by Microtec Research so that you can test the XRAY debugger installation independently from the connection to the emulator.

How to get the Microtec Research HP XRAY ICE for C/C++ product

The HP connection software cannot be used without first installing the Microtec Research HP XRAY ICE product. To order software from Microtec Research:

North America: telephone (408) 980-1300 or (800) 950-5554
　　Microtec Research, Inc.
　　2350 Mission College Blvd.
　　Santa Clara, CA 95054
United Kingdom, Spain, Portugal, Italy, Israel, India, and South Africa: telephone +44 1256 707784
　　Microtec Research Ltd.
　　8 Elmwood
　　Chineham Business Park
　　Basingstoke
　　RG24 8WG
　　England
Germany, Austria, and Switzerland: telephone (49) 089 / 609 00 81
　　Microtec Research GmbH
　　Haidgraben 1 c
　　D-85521 Ottobrunn/Munich
　　Germany
Scandinavia and Finland: telephone +46-8-744-53-00
　　Microtec Research AB
　　Årstaängsvägen 17
　　S-117 43 Stockholm
　　Sweden

France and Benelux: telephone (33) 1-30-12-02-10
　　Microtec Research S.A.
　　Le Sesame
　　8 rue Germain Soufflot
　　78180 Montigny Le Bretonneux
　　France
Japan: telephone (81) 3-5210-3050
　　Nihon Microtec Research, K.K
　　Sanbancho MS Building
　　20 Sanbancho Chiyoda-ku
　　Tokyo 102
　　Japan
Korea: telephone (82) 2-517-0838
　　Hankuk Microtec Research
　　#302 Kyunghee Building
　　109-18 Samsung-dong, Kangnam-ku
　　Seoul, Korea
　　135-090
China: telephone 10-2022744
　　China Microtec Research
　　Golden Bridge Hotel, Room #328
　　38 Bi San Huan Zhong Lu Bei Tai
　　Ping Zhuang, Beijing China 100088

Setting Environment Variables

Set the following variables in the environment of each user:

Variable	What to set it to	Default
USR_MRI	Installation directory for Microtec Research software	/usr/mri
XRAYMASTER	Installation directory for XRAY MasterWorks software	$USR_MRI/master
LD_LIBRARY_PATH	(SunOS/Solaris) Add $XRAYMASTER/lib (location of shared libraries)	none
SHLIB_PATH	(HP-UX) If software is not loaded in the default location, set to $XRAYMASTER/lib:$HP64000/master/shinsei/objects	none
LM_LICENSE_FILE	Location of the license file	/usr/local/flexlm/ licenses/license.dat
HP64000	Installation directory for HP software	/usr/hp64000
PATH	Add $USR_MRI/bin, $XRAYMASTER/bin, and $HP64000/bin	existing $PATH

For Sun systems, $LD_LIBRARY_PATH *must* be set. This variable does not need to be set for HP-UX systems.

If you install all of the software in the default directories, setting the other variables is optional, but recommended. The examples in this and other manuals assume that the variables have been set, so if you don't set the variables, you'll need to memorize the default directories.

Examples

To set the variables for a user who uses the ksh shell on a HP-UX system, the following lines could be placed in the user's .profile file:

```
export USR_MRI=/usr/mri
export XRAYMASTER=$USR_MRI/master
export LM_LICENSE_FILE=/usr/local/flexlm/licenses/license.dat
export HP64000=/usr/hp64000
export PATH=$PATH:$USR_MRI/bin:$XRAYMASTER/bin:$HP64000/bin
```

To set the variables for a user who uses the csh shell on a Sun system, the following lines could be placed in the user's .login file:

```
setenv USR_MRI /usr/mri
setenv XRAYMASTER $USR_MRI/master
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$XRAYMASTER/lib
setenv LM_LICENSE_FILE /usr/local/flexlm/licenses/license.dat
setenv HP64000 /usr/hp64000
setenv PATH $PATH:$USR_MRI/bin:$XRAYMASTER/bin:$HP64000/bin
```

Installing the HP Connection Software

Requirements

You will need the following manuals to complete the installation:

- Microtec Research *Product Installation (UNIX Hosts)*
- Microtec Research *Integrated Environment Utilities*
- *HP 64000-UX for Solaris and SunOS Software Installation Guide* (for SunOS or Solaris hosts)
- *HP 64000-UX for HP-UX 10.xx Software Installation Guide* (for HP-UX 10.xx hosts)

Before you install the HP Connection for Microtec Research MasterWorks XRAY HP ICE (called the "connection software" in this section):

- Check that the Microtec Research XRAY HP ICE software has been installed and that the required environment variables have been set. (See page 13 for installation instructions.)

The connection software must be installed *after* the Microtec Research XRAY HP ICE software..

Microtec Research software version compatibility

- Assembler ASM68K version 7.1 C or later
- C Compiler MCC68K version 4.5 G or later
- C++ Compiler CC68K version 2.0 F or later
- XRAY ICE for C version 3.6 J
- XRAY ICE for C++ version 3.6 J
- XRAY Master version 1.3 E or later

SunOS/Solaris operating system compatibility

- SunOS version 4.1.3 or later
- Solaris version 5.3 or later, including Generic_101945 patch

To verify that the "Generic_101945" patch has been installed, type"

```
uname -a
```

The output should look like:

```
SunOS machine_name 5.4 Generic_101945-10 sun4c sparc
```

HP-UX operating system compatibility

The connection software is compatible with HP 9000 Series 700 workstations running HP-UX version 9.0 and later, with Motif 1.2 (X11R5).

For HP-UX 9.01, 9.03, 9.05, or 9.07, the following patches are required:

- PHSS_5653 or PHSS_6984
- PHSS_5568 (not required for HP-UX 9.07)
- PHCO_4583

Emulator compatibility

Processor	Emulator	Emulator/Analyzer Interface
68000, 68EC000, 68HC000, 68HC001	HP 64742 HP 64743 HP 64744	HP B1468
68302	HP 64746 HP 64798C	HP B1469 HP B3093
68EN302	HP 64798E	HP B3093
68LC302	HP 64798F	HP B3093
68020, 68EC020	HP 64748	HP B1475
68030, 68EC030	HP 64747	HP B1479
68040, 68EC040, 68LC040	HP 64783	HP B3090
68331, 68332, 68334, 68336, 68338	HP 64782	HP B3092
68340	HP 64751	HP B1442
68360, 68EN360	HP 64780	HP B3091
all CPU32 (6833x, 68360, 68340, 68370, 68376, 68396)	HP E3490A or HP E3458A Processor Probe	not available

Note that the connection software is *not* compatible with the HP 64749 emulator.

The connection software is compatible with version 7.0 and later of the HP emulator/analyzer graphical interface software.

Early versions of the HP E3490A Processor Probe were called "HP E3490A Software Probe."

Step 1: Install the Microtec Research XRAY HP ICE software

1 Follow the instructions in the *Microtec Research Product Installation (UNIX Hosts)* document.

Note that the tape containing XRAYMASTER should be loaded first.

2 When the list of features to load from the tape is presented, look at the XHS68K, XRAYMASTER, and LICENSE features. If any of these features is set to "N", change the "N" to a "Y" and continue the installation.

Remember to set the environment variables, as described on page 8.

See also   "Defining Environment Variables" in the "Getting Started" chapter of the Microtec Research *Integrated Environment Utilities* manual.

"Setting Environment Variables" in the "Running the Software" chapter of the Microtec Research *Product Installation* manual.

Step 2: Verify the XRAY MasterWorks installation

1 Start the MasterWorks control panel:

```
MasterWorks
```

2 Double-click the xhs68k icon to start the debugger simulator.

3 Load a file and try a few commands.

A temporary license for XHS68K is provided with the XRAY debugger to help you check that the software has been properly installed. The temporary license will expire on the date listed on the License Authorization Form provided by Microtec Research.

See also   If you have not used XRAY MasterWorks before, now is a good time to go through the tutorial in the Microtec Research *XRAY Debugger Reference Manual.* Later in this chapter, you'll go through the tutorial again, this time connected to an emulator.

Step 3: Install the HP software

- Follow the instructions in the *Software Installation Guide* which accompanies the software media.

  The software will be installed in /usr/hp64000/master by default. Remember to set the HP64000 environment variable to /usr/hp64000 or to the directory under which you installed the software.

  The HP connection software will be installed, and the HP emulator/analyzer graphical interface software (if applicable) will be installed or updated.

Note   The combination of the Microtec Research MasterWorks XRAY HP ICE Debugger and the HP connection software that you have just installed will be referred to as the "XRAY debugger" throughout the rest of this manual.

Step 4: Add the HP notebook pages

1 Start Masterworks

    MasterWorks

2 Double-click the **Defaults** icon.

3 In the MasterWorks Defaults Editor window, click the **MWX** button.

4 Click on the **Position** tab.

5 Scroll down until you see the **User Supplied Notebooks** section.

  If you do not see this section, and you are using HP-UX 9.x, make sure you have installed the necessary patches (see page 11).

6 Enter the full path to the HP dialang file:

    /usr/mri/master/dialang/hp.dia

  This file defines the HP notebook pages which are added to the debugger interface.

  If you installed the HP software in a non-default location, enter $XRAYMASTER/dialang/hp.dia (enter the value of $XRAYMASTER; this dialog does not expand shell variables).

At this point, the Defaults Editor should look like this:

7 Click **Apply All.**

8 You will be prompted to save the default information in a file with a .def extension. Enter:

```
$HOME/.MasterWorks/master.def
```

See the Microtec Research *Integrated Environment Utilities* manual for more information on XRAY MasterWorks defaults files and search paths.

9 Click **Close**.

10 Exit the MasterWorks Defaults Editor by selecting **File→Exit**.

Note	Adding the HP notebook pages to the debugger interface will also add them to the **xhs68k** simulator interface. Using the HP notebook pages in the **xhs68k** interface will result in an error message about an invalid option. These options are only valid in the **xhi68khp** interface.

Step 5: Verify the HP software installation

- If you are using an emulator, start the HP emulator/analyzer graphical interface:

  `emul700 emulator_name`

  where *emulator_name* is the LAN address or logical name of your emulator.

  When the interface has started, exit by selecting **File→Exit→Released**.

- If you are using an HP E3490A Processor Probe, check for the existence of the directory $HP64000/inst/emul/E3490A.

See also   The HP emulator/analyzer graphical interface *User's Guide* for your emulator.

Step 6: Verify the installation of the HP notebook pages

1   From the XRAY debugger interface, pull down the **Notebooks** menu.

2   Check that the last two selections are:

HP Misc Commands

HP Flash Commands

If you need to reinstall software

The connection software makes changes to the Microtec Research XRAY HP ICE product. If you update or reinstall the MRI software, you should run the connection software installation script:

1 Log in as root.

2 Set the environment variables listed on page 8 and the HP64000 environment variable.

3 Change to the directory which holds the installation script:

`cd $HP64000/bin`   (for SunOS or Solaris)

`cd /system/64700MWX`   (for HP-UX 9.x, reinstalling from tape)

`cd $HP64000/system/64700MWX`   (for HP-UX 10.x, or HP-UX 9.x CD-ROM)

4 Run the installation script. Enter:

`envinstall`   (for SunOS or Solaris)

`customize`   (for HP-UX)

Using the XRAY Debugger Tutorial

To help you verify the installation and get comfortable using the XRAY debugger connected to an emulator (or HP Processor Probe), a customized version of the Microtec Research XRAY debugger tutorial program is supplied with the HP connection software.

The *XRAY Debugger Reference Manual* gives additional explanations and pictures, which are not included here.

If you have problems

See the "Solving Problems" chapter beginning on page 81

Before you begin the tutorial

1 Export the following shell variables if they are not already set:

export USR_MRI=/usr/mri (or the directory where you installed the Microtec Research software)

export MRI_68K_BIN=$USR_MRI/bin

export HP64000=/usr/hp64000 (or the directory where you installed the HP software)

export PATH=$PATH:$HP64000/bin

2 Check that the emulator or HP E3490A Software probe is available:

emul700 -lv *emulator_name*

If the emulator is locked:

- If it is locked, you can unlock it with the command:

  emul700 -U *emulator_name*

18

- If it is running via another user, ask that user to release the emulator.

- You can also unlock the emulator by cycling power to the emulator.
  Remember that your target should be powered off first and then the
  emulator powered off, then the emulator powered up and finally the target
  powered up. If you are using the demo board supplied with this emulator
  you need not worry about the sequence of powering the target.

Set Up the Tutorial

Step 1: Copy the Tutorial Files

Substep 1 Create and change to a working directory:

```
mkdir mydemo
cd mydemo
```

2,3,4 A script is supplied which executes substeps 2,3,and 4. To run this
script, enter:

```
$HP64000/demo/master/mwdemo_setup
```

This script also installs configuration files which should be used with the
demo and opens the MasterWorks Control Panel. Later on in the demo, you
will need to know the name of the configuration file to be loaded for the
emulator you are connecting to. For example, the appropriate file for a 68360
emulator is Config68360.EA. Other installed config files are named similarly
based upon the processor type.

If you are using an HP 64798 emulator, use the configuration file
Config6830x.EA. If you are using an HP 64746 emulator, use Config68302.EA.

If you are using the Processor Probe with a 68332 BCC demo card, use the
configuration file ConfigE3490.EA. If you are using the Software Probe with
any other target you will need to define a configuration file for the target.

For some target systems, you may need to re-compile the demo to match the
target RAM location.

Step 2: Build the Program

Substep 1 Invoke a shell window. In the shell window, compile the mw_otis.c
program provided with your distribution files by entering the
following:

For emulator connections:

```
mcc68k -g -Gf mw_otis.c
```

For Software Performance Analyzer (SPA) with emulator connections:

```
mcc68k -g -Gf -Oe -Kt mw_otis.c
```

The compiler options used here are:

- -g   "Full Debug Information"
- -Gf "Pathnames for Source"
- -Oe "Generate Only One Exit From a Function"
- -Kt "Generate Entry/Exit Tags"

For HP E3490A Processor Probe connections with a 68332 BCC demo card:

```
make
```

The make uses the Makefile which was installed by the mwdemo_setup script. It contains compiler options along with linker options necessary to compile and link the demo into the target memory space. If you are using a different target, you will need to modify the Linkcom.cmd file and the configuration file (ConfigE3490.EA) to properly define your target system.

Note that the program will not compile correctly the first time. Substeps 2 and 3 will correct the problem.

2 Correct the error in the source file mw_otis.c by removing the following line:

```
ERROR_delete_this_line
```

3 Recompile mw_otis.c by using the same command as before.

Run the Program

Step 1: Invoke XRAY Debugger

Substep 1   Click the **xhi68khp** icon in the MasterWorks Control Panel.

The MasterWorks Control Panel was opened by the mwdemo_setup script (page 19).

2 Select **Start With Arguments...**.

This will open a dialog box so you an enter the necessary arguments.

3 Enter **-e** followed by a space, the lan name or lan address of your emulator or Processor Probe, a comma, and the name of the configuration file.

For example, if your emulator is "emul360" and the configuration file you found (see page 19) is "Config68360.EA", enter

```
-e emul360,Config68360.EA
```

Wait until you see the message in the log window that the configuration file you specified was loaded before continuing. (If necessary, resize or scroll the Code window.)

See page 35 for other ways to start the debugger.

If you see an error message, see page 82.

Step 2: Load the Executable and Include File

1 Select any of the **Debugger Files** options of the **Notebooks** menu of the Code window.

2 Click the **Load** tab.

3 Click the file folder icon and select **File Chooser** from the menu.

4 Click the button for the **mw_otis.x** file.

5 Click **OK** in the File Chooser.

6 Click the **Load** button below the notebook.

The file mw_otis.x is loaded into the target system.

7 Click the **Incl** tab.

8 Click the file folder icon and select **File Chooser** from the menu.

9 Select **step1.inc** and click **OK**.

10 Click the **Include** button below the notebook.

The include file contents are executed. Two elevator window frames appear.

11 Organize the windows by closing the notebook (click the **Close** button below the notebook) and moving the elevator windows so that both are visible.

Step 3: Single-Step Through the Program

In this step, you will execute the program one line at a time; the code displayed in the Code window follows the program counter.

1 Click **Step** from the debugger window.

The rectangular cursor in the Code window moves to the next executable line of code.

2 Click the **Assembly** button at the lower left of the Code window to view the code in assembly mode.

3 Click **Step** from the debugger window.

4 Step through program execution by clicking the **Step** and **StepI** (step instruction) buttons.

5 Click the **Source** button at the lower left of the debugger Code window to return to high-level mode.

Step 4: Set Up Breakpoints

In this step, you will load an include file that sets up a series of execution breakpoints in the source code. When execution reaches any of these breakpoints, the debugger activates a macro attached to the breakpoint. These breakpoints update the user-defined windows containing elevator displays to simulate the progress of the hardware.

1 Open the Debugger Files notebook by selecting any of the **Debugger Files** options of the **Notebooks** menu.

2 Click the **Incl** tab.

3 Click the file folder icon and select **File Chooser**.

4 Select **step2.inc** in the File Chooser and click **OK**.

5 Click **Include**.

6 Click **Close** in the notebook.

Step 5: Enter Input

1 Select **Restart** from the **File** menu in any debugger window.

2 In the Code window, scroll to line 135.

```
135 if (target_floor != -1)
```

Select the line by highlighting some text in that line.

3 Click the **GoUntil** button to advance program execution to this line.

4 Click in the command entry field in the Code window (following the **>** prompt) and enter:

```
up_button(2)
```

Press the Return key to enter the function call.

5 Similarly, enter:

```
up_button(4)
down_button(3)
elevator2_button(4)
elevator1_button(5)
```

These function calls simulate input to the elevator program.

Step 6: Execute the Program

In this step, you will begin program execution. The function calls entered in the previous step have been queued and simulate input to the system when you execute the program.

1 Click the **Go** button in any debugger window.

2 Wait for the last elevator to reach its final floor and open its doors, then click the **Stop** button.

The normal operation of the elevator would have the program continue in a wait state until it received more input. Because you are simulating the input, you must halt program execution in order to produce the input.

The elevator program will continue running in a "wait state" unless you manually interrupt its operation. You must stop the elevator to continue with the next step.

Step 7: Change a Program Data Object

In this step, you will use the Inspector window to view and modify the contents of an element of a structure defined in the target program.

1 Select **Inspector** from a **Displays** menu in any debugger window.

2 Enter the symbol name **elevator** followed by a return in the Inspect Symbol Name or Expression text field at the bottom of the Inspector window. (You can also select elevator in the Code window and click the **Inspect** button.)

The data object elevator is displayed in the Inspector window. The symbol is an array of structures, so only the address is shown in the value field.

3 Click the expand (**+**) button to the left of elevator.

4 Click the expand (**+**) button of element **[00]** of elevator.

5 Click the expand (**+**) button of the member **button_light** to display its elements.

6 Click and hold the right mouse button in the Value field of element **[00]** of **button_light**, and select **Set Value** from the menu that appears.

7 Change the value in the dialog box from **OFF** to **ON** and then click the **Set** button.

A state is set in the target program that is equivalent to pressing the ground floor button in the elevator.

8 Select **Close Window** from the **File** menu to close the Inspector window.

Step 8: Make a Change in the Source Code

In this step, you will edit the source file, save the source file, and then rebuild the executable. The modification operates the elevator system with only one elevator.

Substep 1 Make the Code window visible.

You can open or raise a debugger window by selecting the window type from the Displays menu in any other debugger window.

2 Type **#1** followed by a return in the Display text field in the Code window to move the code display to line 1 in the source program.

3 Select **Text Editor** from the **Edit File** pull-right menu in the **Edit** menu of the Code window.

The XRAY MasterWorks File Editor opens.

4 Change the number of elevators defined by changing the appropriate #define line to:

```
#define NO_OF_ELEVATORS 1
```

5 Select **Save** from the **File** menu in the editor.

6 Select **Exit** from the **File** menu in the editor.

7 In a shell window, recompile the mw_otis.c program.

Enter the same compile command that you used in Step 2: Build the Program from under the heading "Set Up the Tutorial".

8 Select the **Load** option from the **File** menu in a debugger window to reload the modified mw_otis.x executable.

Step 9: Run the Program to Verify the Effects of Your Changes

The change you made in Step 8 changes the number of elevators enabled, but two elevators will still be displayed (the macros in step1.inc set up the elevator windows). In these steps you will reset the elevator system by clearing the breakpoints and resetting them, and you will verify that the new executable defines only one elevator.

1   In the Code window command entry field, enter **clear** followed by a return to clear all of the breakpoints.

2   Include the file step2.inc again by opening the Debugger Files notebook (**Notebooks** menu), clicking the **Incl** tab, and then clicking the **Include** button. The contents of the page are the same as when you entered the file name previously.

3   Click the **Close** button in the notebook.

4   Choose **Restart** from the **File** menu in any debugger window;.

5   Click the **Step** button to enter the main function.

6   Enter the following function calls in the command entry field in the Code window to demonstrate that only one elevator is defined:

```
up_button(4)
down_button(3)
```

7   Click the **Go** button in any debugger window.

With two elevators operational, one elevator would move to the fourth floor and the other would move to the third. Now, because you modified the source code, the second elevator does not react. The first elevator moves to the third floor, opens, closes, and continues to the fourth floor.

8   Click the **Stop** button after the elevator stops at the fourth floor and opens its doors.

Step 10: Exit XRAY MasterWorks

To experiment with the HP emulator/analyzer graphical interface or the HP Software Performance Analyzer interface, you may continue with the next section before exiting XRAY MasterWorks. To exit now:

1   Select **Exit Masterworks** from the **File** menu of the XRAY MasterWorks Control Panel.

2   Click the **Yes** button.

Continue the Tutorial Program with the HP Emulator/ Analyzer Interface

Continue the tutorial with these steps to try out the HP emulator/analyzer graphical interface. Many more commands and capabilities are described in the emulator/analyzer graphical interface *User's Guide.*

Step 1: Clear all the Breakpoints

- In the command entry field in the XRAY debugger Code window, enter "Clear".

Step 2: Start the Emulator/Analyzer

- Enter this command in the XRAY debugger command entry field:

  ```
  ss emul
  ```

- Or, from the UNIX command line, enter:

  ```
  emul700 emulator_name
  ```

Step 3: Let the Program Loop

```
Do this by entering "go" in the command entry field in the
XRAY debugger Code window.
```

Step 4: Experiment with some of the Emulator/Analyzer Commands

See the emulator/analyzer graphical interface *User's Guide* for information on how to enter commands.

Here are a few commands which use the analyzer's trace capability:

1 Select **Trace→Everything**.

This presents a default trace of what is happening on the analyzer bus.

2 Enter "select_elevator_button" in the Entry buffer (): .

3 Select **Trace→About( )**

This shows program flow before and after a call of select_elevator_button().

4 Enter "range elevator thru elevator end" in the Entry buffer () : .

5 Select **Trace→Only( )**

This shows all memory accesses to the elevator data structure.

Step 5: Exit or Continue with the Software Performance Analyzer

- To exit now, exit from the Emulator/Analyzer interface, the XRAY debugger interface or the MasterWorks Control Panel. Select the Exit option from under the File pull down menu.

Continue the Tutorial Program with the HP Software Performance Analyzer

Continue the tutorial with these steps to try out the HP Software Performance Analyzer interface. The HP Software Performance Analyzer requires an HP 64700-series emulator; if you are using an HP Processor Probe, skip to the next section. For more information, refer to the *HP B1487 Software Performance Analyzer User's Guide*.

Step 1: Compile and Load the Program

If the demo program has not been compiled as described on page 20, then recompile the program with the appropriate options for the HP Software Performance Analyzer:

1 Click **Stop**, then click **Clear** to clear all breakpoints.

2 Enter the following command on the UNIX command line:

```
mcc68k -g -Gf -Oe -Kt mw_otis.c
```

3 Load the new executable file into the debugger.

Step 2: Clear all the Breakpoints

- In the command entry field in the XRAY debugger Code window, enter "Clear".

Step 3: Start the Software Performance Analyzer

1 On the UNIX command line, set the shell variable:

```
export HPSPAMARKERS="yes _r_%f_ _r_%f_+2"
```

This environment variable informs the HP Software Performance Analyzer to use the MRI function tags at the beginning and end of every function.

2 From the UNIX command line, enter:

```
emul700 -u xperf emulator_name
```

Step 4: Let the Program loop

- Do this by entering "go" in the command entry field in the XRAY debugger Code window.

Step 5: Experiment with some of the HP Software Performance Analyzer Commands

See the *HP B1487 Software Performance Analyzer User's Guide* for information on how to enter commands into the user interface. Here are a few profile commands you can try:

1 Select **Program Activity**.

This shows where the processor is spending most of its cycles. Ranges could be defined to cover entire libraries.

2 Select **Function Duration**, then select **Table**.

This shows the number of times each of the functions is called and its maximum, minimum, and mean duration.

Function duration in this measurement includes function calls. Function duration measurements are made using the Microtec Research tags that are compiled into each function. One tag is written at function entry, and a second tag is written at function exit. The HP Software Performance Analyzer refers to these tags as *markers*.

3 Select **Profile→Profile...**, select **Function Duration Excluding Profiled Calls**, then click **OK**.

This measurement is similar to the preceding one, except that function duration excludes calls to other functions that are being measured.

4 Position the cursor on one of the functions displayed. Press the right mouse button and select **Expand Event Toggle**.

This breaks the function into more time ranges.

Step 6: Exit or Continue with the Branch Validator

- To exit now, select **File→Exit.**

  You can exit from the HP Software Performance Analyzer interface, the XRAY debugger interface, or the MasterWorks Control Panel.

Continue the Tutorial Program with the HP Branch Validator

Continue the tutorial with these steps to try out the HP Branch Validator (BBA) interface.

Step 1: Compile and Load the Program

1 Click **Stop**, then click **Clear** to clear all breakpoints.

2 Compile mw_otis.c so that the executable will contain branch information.

If you are using an emulator, enter the following command on the UNIX command line:

```
mcc68kbba -g -Gf mw_otis.c
```

If you are using an HP E3490A Processor Probe, the compilation command is included in the Makefile. From the UNIX command line, enter:

```
make clean
make mw_otis.bba
```

3 Load the new executable file into the debugger.

Step 2: "Unload" the HP Branch Validator data

1 Open the **HP Misc Commands** notebook.

2 Select the **BBA** option.

3 In the "BBA file options" section, select **Dump data into default file: bbadump.data**.

The captured data will be saved into file bbadump.data.

4 In the "Add offset to Base Address" section, select **Use Default (0x0)**.

5 Click **Apply**.

This executes the "ss bbaunload DEFAULT DEFAULT" command. A message will appear in the log window to inform you that the BBA unload is complete. The file bbadump.data has been created in the directory from which you are running the debugger. This file will accumulate information each time the bbaunload command is executed.

Step 3: Run the Program

1 Click **Go** and let the program loop for a short time (about 1 second).

2 Click **Stop**.

Step 4: View the BBA Report

1 In the HP Branch Validator - BBA notebook page, click **Apply** to execute the bbaunload command again.

2 On the UNIX command line, enter

```
bbarep
```

This will display a report of how many branches were followed in each function. Here is an example of the report:

```
_hit___total____%____IA___function_____file _____
  0 /    4 (  0.00)   move_elevator           mw_otis.c
  3 /    4 ( 75.00)   main                    mw_otis.c
  2 /    2 (100.00)   init_elevators          mw_otis.c
  7 /    9 ( 77.78)   select_floor_to_go      mw_otis.c
  6 /    9 ( 66.67)   select_any_call_button  mw_otis.c
  0 /    8 (  0.00)   select_call_button      mw_otis.c
  3 /    5 ( 60.00)   select_elevator_button  mw_otis.c
  0 /    3 (  0.00)   push_button             mw_otis.c
  0 /    1 (  0.00)   elevator1_button        mw_otis.c
  0 /    1 (  0.00)   elevator2_button        mw_otis.c
  0 /    2 (  0.00)   up_button               mw_otis.c
  0 /    2 (  0.00)   down_button             mw_otis.c
21 out of 50 branches executed (42.00%)
```

3 Display a more detailed report. On the UNIX command line, enter:

```
bbarep -b2 -a2
```

This will give you information about functions which were never called.

Step 5: Try to Increase BBA Coverage

Try to increase the BBA coverage by calling "elevator1_button" and "elevator2_button":

1 Include step2.inc agian.

This will restore all of the breakpoints.

2 Choose **Restart** from the **File** menu in any debugger window.

3 Scroll down to line 135:

```
135 if (target_floor != -1)
```

4 Highlight some text, then click **GoUntil**.

5 Click in the command entry field in the Code window and enter:

```
up_button(2)
up_button(4)
down_button(3)
elevator2_button(4)
elevator1_button(5)
```

6 Click the **Go** button in any debugger window.

7 Wait for the last elevator to reach its final floor and open its doors, then click the **Stop** button.

8 In the HP Branch Validator - BBA notebook page, click **Apply** to execute the bbaunload command again.

9 Re-enter the **bbarep** command on the UNIX command line.

Notice that the BBA coverage has increased greatly with the new function calls:

```
_hit___total____%____IA____function_____file _____
    4 /    4 (100.00)   move_elevator          mw_otis.c
    4 /    4 (100.00)   main                   mw_otis.c
    2 /    2 (100.00)   init_elevators         mw_otis.c
    9 /    9 (100.00)   select_floor_to_go     mw_otis.c
    8 /    9 ( 88.89)   select_any_call_button mw_otis.c
    5 /    8 ( 62.50)   select_call_button     mw_otis.c
    5 /    5 (100.00)   select_elevator_button mw_otis.c
    3 /    3 (100.00)   push_button            mw_otis.c
    1 /    1 (100.00)   elevator1_button       mw_otis.c
    1 /    1 (100.00)   elevator2_button       mw_otis.c
    2 /    2 (100.00)   up_button              mw_otis.c
    2 /    2 (100.00)   down_button            mw_otis.c
46 out of 50 branches executed (92.00%)
```
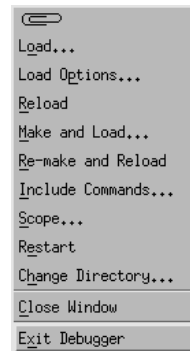
For more information about the HP Branch Validator, see the *HP Branch Validator (BBA) User's Guide*.

Step 6: Exit

You can exit from the XRAY debugger interface or the MasterWorks Control Panel.

- Select **File→Exit.**

```
⊂▭⊃
Load...
Load Options...
Reload
Make and Load...
Re-make and Reload
Include Commands...
Scope...
Restart
Change Directory...
Close Window
Exit Debugger
```

2

Common Tasks

Starting the Debugger

You can start the XRAY debugger from the MasterWorks Control
Panel, or from the UNIX command line.

Waiting for emulator initialization

Before you use the debugger, the debugger must initialize the
emulator. As the XRAY debugger starts, you should see the following
three messages in the debugger window:

```
NOTE: Connecting to emulator: emulator_name
NOTE: Connected!
NOTE: Loaded configuration file: config_filename
```

The first message tells you the name of the emulator for which a
connection has been requested.

The second message informs you that a successful connection has
been made.

The third message informs you that a configuration has been loaded.
This configuration may be the default configuration (if you have just
started a new session with the emulator) or a specific configuration (if
you are restarting the debugger after having previously exited a
session, leaving the emulator locked). See page 43 for information on
how to leave the emulator locked when exiting.

In addition, if you have specified a configuration file to be loaded by
requesting one in the command line arguments, a fourth message will
be displayed, showing the name of the loaded configuration file.

N o t e Do not attempt to perform any tasks in the debugger until all of these
messages have appeared. If necessary, resize the Code window so that
several lines of the output display are visible.

To start the XRAY debugger from the MasterWorks Control Panel
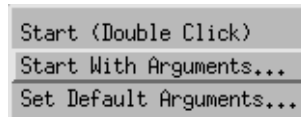
1 Start the MasterWorks control panel:

`MasterWorks`

Note that MasterWorks should not be run as a background process.
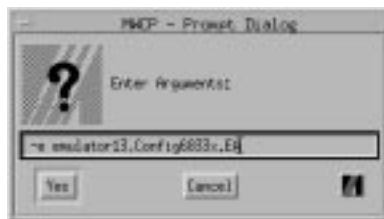
2 Click the xhi68khp icon.

3 Select **Start With Arguments...**.

This will open a dialog box so you an enter the necessary arguments.

4 Enter **-e** followed by the name of your emulator, immediately followed by a comma and the name of a configuration file (if any).

The emulator name is required. The configuration file name is optional; if you don't specify a configuration file, a default configuration will be loaded.

Note that there is no space between the emulator name, the comma, and the configuration file name.

See also   Page 63 for more information about loading configurations.

To set default arguments for the XRAY debugger

To avoid entering the emulator name, configuration file, and command-line options every time you start the XRAY debugger, you can enter the arguments once and save them as default arguments:

1   Choose **Set Default Arguments...** from the xhi68khp icon menu.

2   Enter the arguments.

The default arguments will be used whenever you double-click the xhi68khp icon, or when you choose **Start (Double Click)** from the xhi68khp icon menu.

To start the XRAY debugger from the MasterWorks Control Panel with default arguments

1   Start the MasterWorks control panel:

MasterWorks

2   Double-click the xhi68khp icon.

To start the XRAY debugger from the UNIX command line

- Type xhi68khp, followed by -e and your emulator name, immediately followed by a comma and the name of a configuration file (if any).

  The emulator name is required.

  The configuration file name is optional; if you don't specify a configuration file, a default configuration will be loaded. This default configuration is supplied in the appropriate directory under $HP64000/inst/emul and should never be modified.

Example

If your emulator is named "hpemul332" and you have a configuration file named "Config.EA", type:

```
xhi68khp -e hpemul332,Config.EA
```

See also

To use debugger command-line options

- You can specify other command-line options when you start the XRAY debugger. Enter these options:

  - when you start the debugger from the UNIX prompt

  - when you use Start With Arguments from the MasterWorks Control Panel, or

  - as default arguments.

  options can be placed before or after the -e option, but no options may be placed between the emulator name and the configuration file name.

Command-line options for xhi68khp

-e *emulator* The -e option tells the XRAY debugger which emulator it should connect to. This option is required. You may also specify the name of a configuration file to load. If you do not specify a configuration file, the default configuration will be loaded.

The usual options for the XRAY debugger are also supported:

[*abs_file*] Loads an absolute (.x) file.

[-b] Run the XRAY debugger in the background.

[-ni] No image—load symbols only.

[-i *file*] Runs commands from an include file (command file).

[-j *file*] Specifies a journal file into which debugger commands are recorded.

[-l *file*] Specifies a log file into which debugger commands and errors are recorded.

A typical command line might look like this:

```
xhi68khp -e emulator1,Config.EA -i debugcmds -l logfile
```

Starting Other HP Interfaces

If you are using an HP 64700-series emulator, you can start the HP emulator/analyzer and Software Performance Analyzer graphical user interfaces from the XRAY debugger, or from the UNIX command line.

Use the XRAY debugger to enter all commands, and go to the HP interfaces only when you need special capabilities such as trace analysis.

To start the HP emulator/analyzer graphical user interface from the XRAY debugger

1 Switch to the XRAY debugger Code window.
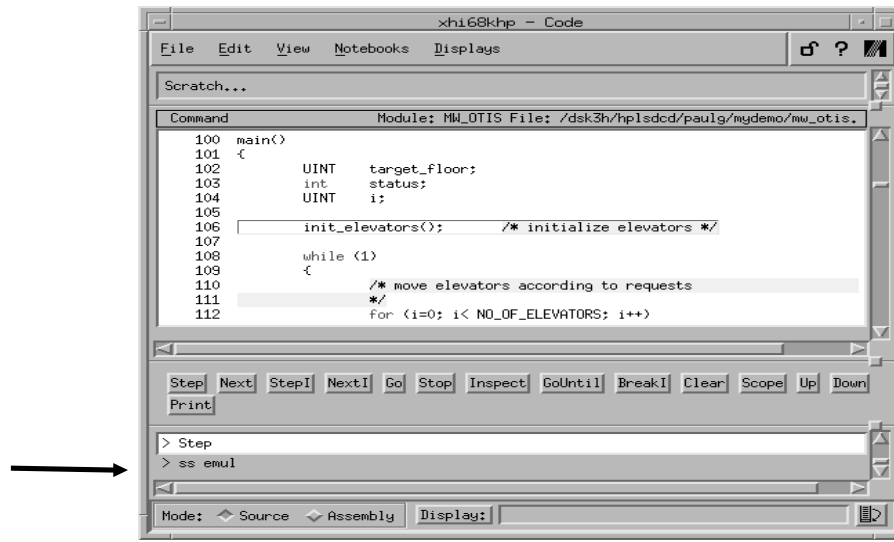
2 In the command entry field, type:

```
setstatus emul
```

This command may also be entered in its abbreviated form:

```
ss emul
```

You don't need to enter the name of an emulator, because the XRAY debugger already knows which emulator it is connected to.

The ss command "emul" is only available for emulator connections; it is not available for the HP E3490A Processor Probe.

```
┌─────────────────────────────────────────────────────────────┐
│ ─                    xhi68khp — Code                    ▵ ▫  │
│  File   Edit   View   Notebooks   Displays         ⌂  ?  ▨  │
│ ┌───────────────────────────────────────────────────────┐ ▲ │
│ │ Scratch...                                            │ ▽ │
│ ├───────────────────────────────────────────────────────┤   │
│ │ Command          Module: MW_OTIS File: /dsk3h/hplsdcd/paulg/mydemo/mw_otis. │
│ │    100  main()                                        │ △ │
│ │    101  {                                             │   │
│ │    102      UINT    target_floor;                     │   │
│ │    103      int     status;                           │ ▬ │
│ │    104      UINT    i;                                 │   │
│ │    105                                                │   │
│ │    106      init_elevators();      /* initialize elevators */│
│ │    107                                                │   │
│ │    108      while (1)                                 │   │
│ │    109      {                                         │   │
│ │    110          /* move elevators according to requests │   │
│ │    111          */                                    │   │
│ │    112          for (i=0; i< NO_OF_ELEVATORS; i++)    │ ▽ │
│ │ ◁                                                    ▷ │   │
│ ├───────────────────────────────────────────────────────┤   │
│ │ Step  Next  StepI  NextI  Go  Stop  Inspect  GoUntil  BreakI  Clear  Scope  Up  Down │
│ │ Print                                                 │   │
│ ├───────────────────────────────────────────────────────┤ ▲ │
│ │ > Step                                                │ ▵ │
│ │ > ss emul                                             │ ▽ │
│ │ ◁                                                    ▷ │   │
│ ├───────────────────────────────────────────────────────┤ ▽ │
│ │ Mode: ◇ Source  ◇ Assembly   Display: |               ▤▷│
│ └───────────────────────────────────────────────────────┘   │
└─────────────────────────────────────────────────────────────┘
```

To start the HP emulator/analyzer graphical user interface from the UNIX command line

• At the operating system prompt, type:

`emul700` *emulator_name*

If you have just started the XRAY debugger (in a script, for example), wait a few seconds before you start the HP interface. If both interfaces try to connect to the emulator at the same time, an error can occur which states that a configuration process is already running.

See also   Your HP emulator/analyzer graphical interface *User's Guide* for information on other command-line options.

To start the HP Software Performance Analyzer interface from the XRAY debugger

1 Switch to the XRAY debugger Code window.

2 In the command entry field, type:

```
setstatus spa
```

This command may also be entered in one of its abbreviated forms:

```
ss spa
```

You don't need to enter the name of an emulator, because the XRAY debugger already knows which emulator it is connected to.

The HP Software Performance Analyzer requires an emulator with the Software Performance analyzer circuit board and the appropriate software.

The ss spa command is not available for the HP E3490A Processor Probe.

To start the HP Software Performance Analyzer interface from the UNIX command line

• At the operating system prompt, type:

```
emul700 -u xperf emulator_name
```

To use the HP Branch Validator (BBA)

You can capture branch coverage data from within the debugger interface if you have the HP Branch Validator (BBA).

1 Compile the program using the **mcc68kbba** command.

mc68kbba is a script which runs bbacpp before compiling with **mcc68k**. If you are using a Makefile, change CC=mcc68k to CC=mcc68kbba.

2 Load the program into the debugger.

3 Select **Notebooks→HP Misc Commands→BBA…,** select any desired options, then click **Apply**.

If you have forgotten to compile the file with mcc68kbba, you will see the error message, "WARNING (HP): No BBA symbols found!"

4 Execute the program.

5 On the UNIX command line, use the **bbarep** command to display a coverage report.

To store BBA data using a setstatus command

In the command entry field, enter:

```
ss bbaunload file offset
```

The parameters are:

file **DEFAULT** stores data in the file bbadump.data
filename stores data in the named file

offset **DEFAULT** uses an offset of 0x0
OFFSET *hex_value* uses the specified hex value as an offset

Example

To dump the captured data in the file mw_otis.dump and use no offset, enter:

```
ss bbaunload mw_otis.data DEFAULT
```

See also

The *HP Branch Validator (BBA) User's Guide* for more information on the **bbarep** command and other features of BBA.

The HP Branch Validator section in the "Getting Started" chapter (page 28).

Exiting the XRAY Debugger

To exit

There several ways to exit the XRAY debugger or the HP interfaces:

- From the XRAY debugger, select **File→Exit Debugger**.

  This closes the XRAY debugger, HP emulator/analyzer and SPA interface windows and releases the emulator.

- From the MasterWorks Control Panel, select **File→Exit MasterWorks**.

  This closes the XRAY debugger window, the MasterWorks Control Panel, the HP emulator/analyzer window, and the SPA interface window and releases the emulator.

- From the HP emulator/analyzer graphical interface or HP Software Performance Analyzer (SPA) interface, select **File→Exit** then **Window**, **Locked**, or **Released**.

  If you choose Window, other interface windows remain open, and the emulator remains locked. If you choose Locked, other interface windows are closed, but the emulator remains locked. If you choose Released, other interface windows are closed and the emulator is released.

- In the XRAY debugger command entry field, enter

  ```
  ss exit
  ```

  This closes the XRAY debugger window and all of the other interface windows. The emulator remains locked to you.

  Locking the emulator prevents other users from connecting to the emulator. If necessary, the emulator can be unlocked by typing "emul700 -U *emulator_name*" at the UNIX prompt.

Using HP Setstatus Commands

The following setstatus (ss) commands are available for use with the
XRAY debugger. To use these commands, enter them in the command
entry field in the Code window.

ss bbaunload [DEFAULT | *filename*] | [DEFAULT | OFFSET *hex_value*]
Unloads the HP Branch Validator data into a file (which can be a
default file or a specified file) or a file with an offset (the offset can be
default or a specified value).

ss config   Opens the Modify Configuration window.

ss config *file*   Loads the configuration file *file* into the emulator.

ss config_modify   Opens the Modify Configuration window.

ss ef *parameters*   Erases flash memory (see page 59 for details).

ss emul   Starts the HP emulator/analyzer user interface. (Not
available for the HP E3490A Processor Probe.)

ss exit   Closes the debugger window; leaves the emulator locked
to the user.

ss pf *parameters*   Programs flash memory (see page 57 for
details).

ss reset   Resets the target processor.

ss reset go   Runs from reset.

ss help   Displays the HP setstatus commands.

ss spa   Starts the HP Software Performance Analyzer. This command requires that the HP B1487 Software Performance Analyzer software has been installed, and that the Software Performance Analyzer board has been installed in the HP emulator frame. (Not available for the HP E3490A Processor Probe.)

Using Registers

You can access registers in three ways:

Basic registers

"Basic" registers can be accessed through the XRAY debugger
Display→**Register** menu. These registers are listed on page 48.

Supported processor-specific registers

Registers which are used only by certain processors must be accessed
using the `print` and `setreg` commands. Enter these commands in
the command entry field in the XRAY debugger Code window.
Individual bits in the Status register may also be accessed in this way.
These registers are listed on pages 48 through 50.

Other registers

The emulator may support other registers which are not listed in the
tables on the following pages, and thus are not accessible through the
`print` or `setreg` commands. Examples of such register are the
emulator copies of the SIM registers (EMSIM registers) and the
Communication Processor Module (CPM) registers.

To access these registers, use the HP emulator/analyzer graphical
interface. In addition, memory-mapped registers can also be accessed
using the memory read and write commands.

Examples

To print the PEPAR register (the 68360 Port E Pin Assignment Register), enter this command in the Code window command entry field:

```
print @pepar
```

To see the value of the PEPAR register in hexadecimal format, type:

```
print /h @pepar
```

To set the C bit in the Status register, enter this command in the Code window command entry field:

```
setreg @c=0x1
```

To set the PEPAR register, enter:

```
setreg @pepar=0xFF
```

See also

Supported registers

- In the XRAY debugger interface, use the register names from the tables which follow. Register names may be different in the HP emulator/analyzer interface.

Registers which may be accessed from the Display→Register menu

Register Name	68000	68020/ EC020	68030/ EC030	68040/ EC040	68302	68340	68360	68F333	6833x
@PC	■	■	■	■	■	■	■	■	■
@D0-D7	■	■	■	■	■	■	■	■	■
@A0-A7	■	■	■	■	■	■	■	■	■
@USP	■	■	■	■	■	■	■	■	■
@SSP	■				■	■	■	■	■
@MSP		■	■	■					
@ISP		■	■	■					
@SR	■	■	■	■	■	■	■	■	■
@VBR		■	■	■		■	■	■	■
@SFC		■	■	■		■	■	■	■
@DFC		■	■	■		■	■	■	■
@CACR		■	■	■					
@CAAR		■	■						

Status register bits—must be accessed via the Code window command entry field

@C	@M
@X	@T0
@Z	@T1
@N	@T
@V	@I
@CCR	@S

Registers which must be accessed via the command entry field in the Code window

68030	68EC030	68040	68302	
@CRP	@AC0	@DACR0*	@BAR	@SCCM1
@MMUSR	@AC1	@DACR1*	@BCR	@SCCM2
@SRP	@ACUSR	@DTT0	@BR0	@SCCM3
@TC		@DTT1	@BR1	@SCCS1
@TT0		@IACR0*	@BR2	@SCCS2
@TT1		@IACR1*	@BR3	@SCCS3
		@ITT0	@CMR	@SCM1
		@ITT1	@CKCR**	@SCM2
		@MMUSR	@CR	@SCM3
		@SRP	@CSR	@SCON1
		@TC	@DAPR	@SCON2
		@URP	@DSR1	@SCON3
			@DSR2	@SCR
			@DSR3	@SIMASK
			@FCR	@SIMODE
			@GIMR	@SPMODE
			@IMR	@TCN1
			@IPR	@TCN2
			@ISR	@TCR1
			@OR0	@TCR2
			@OR1	@TER1
			@OR2	@TER2
			@OR3	@TMR1
			@PACNT	@TMR2
			@PADAT	@TRR1
			@PADDR	@TRR2
			@PBCNT	@WCN
			@PBDAT	@WRR
			@PBDDR	
			@SAPR	
			@SCCE1	
			@SCCE2	
			@SCCE3	

* Not supported by the 68040/EC040 emulator, but reserved by the XRAY debugger.
** 64798C only.

Registers which must be accessed via the command entry field in the Code window

68EN302		68LC302	
@AR_CNTRL	@OR0	@BAR	@SAPR
@BAR	@OR1	@BCR	@SCCE1
@BCR	@OR2	@BR0	@SCCE2
@BR0	@OR3	@BR1	@SCCM1
@BR1	@PACNT	@BR2	@SCCM2
@BR2	@PADAT	@BR3	@SCCS1
@BR3	@PADDR	@BSR	@SCCS2
@CKCR	@PBCNT	@CMR	@SCM1
@CMR	@PBDAT	@CR	@SCM2
@CR	@PBDDR	@CSR	@SCON1
@CSER0	@PCSR	@DAPR	@SCON2
@CSER1	@SAPR	@DISC	@SCR
@CSER2	@SCCE1	@DSR1	@SIMASK
@CSER3	@SCCE2	@DSR2	@SIMODE
@CSR	@SCCE3	@FCR	@SPMODE
@DAPR	@SCCM1	@GIMR	@TCN1
@DBA0	@SCCM2	@IMR	@TCN2
@DBA1	@SCCM3	@IOMCR	@TCR1
@DCR	@SCCS1	@IPDR	@TCR2
@DRFRSH	@SCCS2	@IPLCR	@TER1
@DSR1	@SCCS3	@IPR	@TER2
@DSR2	@SCM1	@ISR	@TMR1
@DSR3	@SCM2	@IWUCR	@TMR2
@ECNTRL	@SCM3	@LPDCR	@TRR1
@ECNFIG	@SCON1	@OR0	@TRR2
@EDMA	@SCON2	@OR1	@WCN
@EMRBLR	@SCON3	@OR2	@WRR
@ETHER_TEST	@SCR	@OR3	
@FCR	@SIMASK	@PACNT	
@GIMR	@SIMODE	@PADAT	
@IER	@SPMODE	@PADDR	
@IMR	@TCN1	@PBCNT	
@INTR_EVENT	@TCN2	@PBDAT	
@INTR_MASK	@TCR1	@PBDDR	
@IPR	@TCR2	@PITR	
@ISR	@TER1	@PNDAT	
@IVEC	@TER2	@PNDDR	
@MBCTL	@TMR1	@RINGOCR	
@MOBAR	@TMR2	@RINGOEVR	
	@TRR1		
	@TRR2		
	@WCN		
	@WRR		

Registers which must be accessed via the command entry field in the Code window

68340	68F333, 68396		6833x, 68370, 68376		68360
@AVR	@CSBR0	@PEPAR	@CSBR0	@DDRE	@AVR
@CS0ADDR	@CSBR3	@PFIVR	@CSBR1	@DDRF	@BKAR
@CS0MASK	@CSBR5	@PFLVR	@CSBR2	@PEPAR	@BKCR
@CS1ADDR	@CSBR6	@PFPAR	@CSBR3	@PFPAR	@BR0
@CS1MASK	@CSBR7	@PICR	@CSBR4	@PICR	@BR1
@CS2ADDR	@CSBR8	@PITR	@CSBR5	@PITR	@BR2
@CS2MASK	@CSBR9	@PORTA	@CSBR6	@PORTC[†]	@BR3
@CS3ADDR	@CSBR10	@PORTB	@CSBR7	@PORTE	@BR4
@CS3MASK	@CSBRBT	@PORTC	@CSBR8	@PORTF	@BR5
@DDRA	@CSOR0	@PORTE	@CSBR9	@RSR	@BR6
@DDRB	@CSOR3	@PORTF	@CSBR10	@SIM_MCR	@BR7
@MBAR	@CSOR5	@PORTFE	@CSBRBT	@SWSR	@CDVCR
@PICR	@CSOR6	@PORTG	@CSOR0	@SYNCR	@CLKOCR
@PITR	@CSOR7	@PORTH	@CSOR1	@SYPCR	@GMR
@PORTA	@CSOR8	@RSR	@CSOR2		@MBAR
@PORTB	@CSOR9	@SCIMMCR	@CSOR3		@MCR
@PORTB1	@CSOR10	@SWSR	@CSOR4		@MSTAT
@PPARA1	@CSORBT	@SYNCR	@CSOR5		@OR0
@PPARA2	@CSPAR0	@SYPCR	@CSOR6		@OR1
@PPARB	@CSPAR1		@CSOR7		@OR2
@RSR	@DDRAB		@CSOR8		@OR3
@SIM_MCR	@DDRE		@CSOR9		@OR4
@SWIV	@DDRF		@CSOR10		@OR5
@SWSR	@DDRG		@CSBRBT		@OR6
@SYNCR	@DDRH		@CSORBT		@OR7
@SYPCR			@CSPAR0		@PEPAR
			@CSPAR1		@PICR
			@CSPDR[†]		@PITR
					@PLLCR
					@RSR
					@SWIV
					@SWSR
					@SYPCR

* Not supported by the 68040/EC040 emulator, but reserved by the XRAY debugger.

[†] CSPDR is used for emulator connections, and PORTC is used for HP Processor Probe connections.

Tracing Program Execution

Here are some examples of how you can use the HP emulator/analyzer interface to trace program execution.

See also   Your HP emulator/analyzer graphical interface *User's Guide* has detailed information on how set up and analyze traces.

To trace what happens just before the program stops

1 Start the HP emulator/analyzer interface, if necessary.

2 In the HP emulator/analyzer interface, select **Display→Memory→Mnemonic at PC**.

3 In the HP emulator/analyzer interface, select **Trace→Until Stop**.

4 Run your target program.

Use the XRAY debugger to start the program running.

Both the XRAY debugger Code window and the HP emulator/analyzer interface will show the line of code where execution stopped.

5 In the HP emulator/analyzer interface, select **Trace→Stop**.

6 In the HP emulator/analyzer interface, select **Trace→Display**.

This procedure can be useful if your program is failing due to an intermittent bug, or if you want to see what happened just before a breakpoint.

The display should look similar to this:
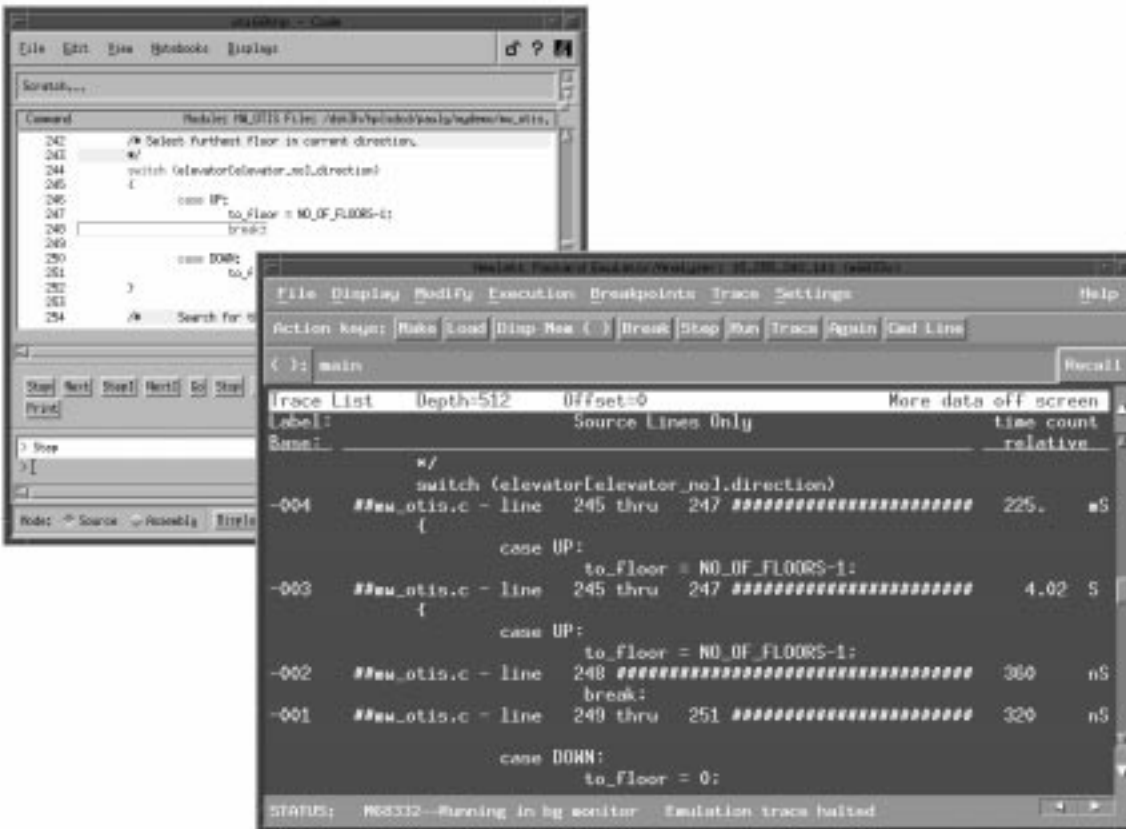
The trace display shows only the lines of code which were actually executed. If code was executed in other functions, that code is displayed in addition to the code in the current function or module.

To trace all accesses to a variable

1 In the HP emulator/analyzer interface, enter the name of the variable in the entry buffer.

The entry buffer is the line beginning with "():".

2 In the HP emulator/analyzer interface, select **Trace**→**Only().**

3 Run your target program.

Use the XRAY debugger start and stop the program.

4 In the HP emulator/analyzer interface, select **Trace**→**Stop**.

5 In the HP emulator/analyzer interface, select **Trace**→**Display**.

To trace accesses to a variable, the analyzer needs to know the address of the variable. Thus automatic (heap) variables cannot be traced.

To trace a variable which covers a range of address locations, enter a description like this:

```
range variable thru variable end
```

into the entry buffer "():".

When specifying a local symbol, include the name of the module. For example, if the symbol "localData" is in "module2.c", specify:

```
module2.c:localData
```

You can also trace function calls using the instructions given above.

See also Your HP emulator/analyzer graphical interface *User's Guide*.

Programming Target Flash Memory

You can use the HP E3490A Processor Probe to program flash memory on your target system.

Target system configuration

Prior to erasing or programming parts, you must to establish any necessary options on the target system. This can include enabling the programming voltage (V_{PP}), disabling write protection, or properly setting the chip selects in the SIM. The chip selects in the SIM can be set through the configuration process. The documentation on the target system can provide information on such things as establishing the V_{PP} and disabling any target based write protection. The HP E3490A Processor Probe accomplishes the FLASH ROM erasure and programming through standard target bus writes and reads.

Supported Parts

The HP E3490A Processor Probe can program standard Intel and AMD parts, and equivalent parts from other manufacturers. Supported algorithms include the Intel quick-pulse (or the identical AMD Flashrite), Intel Auto, AMD 5V embedded, and AMD 12V embedded algorithms.

Supported File Sizes

There is no file size limit.

HP Notebooks

You can program and erase flash parts by using the HP notebook pages or by using the command entry field. It is strongly recommended that you use the HP notebook pages.

Solving Problems

If you see an error message while a flash command is being processed, the flash command was unsuccessful and will need to be repeated with the correct parameters. If problems occur while processing the absolute file, the error condition will be detected by the MRI XRAY debugger. Other errors will be detected by the HP connection software.

Errors detected by the XRAY debugger

If the absolute file does not exist (possibly because of a misspelled name), you will see the message "Unable to open absolute file: **.x", followed by "Terminated when processing absolute file **.x. LOAD /ns **.x". The XRAY debugger will also check that the file is an IEEE-695 format file.

Errors detected by the HP connection software

After the absolute file has been loaded by the XRAY debugger, the HP connection software will check the other parameters of the flash command. Error messages generated by the HP connection software begin with "ERROR (HP)" or "WARNING (HP)". These messages may be displayed in an error dialog box or in the log output window . In either case, the flash command was unsuccessful. You will need to correct the problem and try the command again.

To program a part

1 Create a file which contains the data to be flashed.

2 Set the necessary options (if any) on your target system to allow flash programming.

3 Switch to the XRAY debugger Code window.

4 Set the PC register to a valid value.

When the flash programming has finished, the debugger will read memory at the PC. If the PC points to an invalid address, numerous warning messages will be displayed.

5 From the Notebooks menu, select **HP Flash Commands** then select **Program...**.

If the HP Flash Commands notebook is missing, check that you have installed the HP notebooks as described on page 14.

6 Enter the following information in the Program Flash Command notebook page:

Algorithm: select the type of part to be programmed

Start Address: enter the starting address as a hex value

Size: enter the size (in hex) of the flash ROM in bytes

Bus Width: select the width of the target bus

Device Width: select the width of the flash ROM

File: enter the name of the IEEE-695 file to be loaded

7 Click **Apply**.

To program a part using a setstatus command

Follow steps 1-4 above, then in the command entry field, type:

```
setstatus pf device start_addr ROM_size bus_width
device_width file
```

You must include all of the parameters for this command. The parameters are:

device The programming algorithm used by the flash device:
Basic for Intel quick-pulse (or the identical AMD Flashrite)
Intel for Intel Auto
12V_AMD for AMD 12V
5V_AMD for AMD 5V embedded

start_addr The start address in hex. Example: for a hex address of 10000, enter 0x10000, 010000, or 10000.

ROM_size The size of the ROM in hex. Example: for a size of 8000 bytes (hex), enter 0x8000, 08000, or 8000.

bus_width The width of the bus:
/B for byte
/W for word
/L for long

device_width The width of the device:
/B for byte
/W for word
 /L for long

file The name of the IEEE-695 file to load.

Example

To program a flash ROM with data found in test.x, starting at hex address 100000 with a ROM size of 32768 bytes, a bus width size of 4 bytes, and a device width of 1 byte, enter:

```
ss pf Basic 0x100000 0x8000 /L /B test.x
```

To erase a part

1 From the Notebooks menu, select **HP Flash Commands**, then select **Erase**.

2 Enter the following information in the Erase Flash Command notebook page:

Algorithm: select the type of part to be erased, and whether chip or only selected sectors should be erased

Start Address: enter the starting address as a hex value

Size: enter the size (in hex) of the flash ROM in bytes

Bus Width: select the width of the target bus

Device Width: select the width of the flash ROM

Sectors to Erase: if you selected an algorithm which erases only selected sectors, enter the sector address values in hex

3 Click **Apply**.

After the part has been erased, the following message will be displayed:

```
STATUS (HP): Erase flash command successful.
```

To erase a part using a setstatus command

In the command entry field, type:

```
ss ef device start_addr ROM_size bus_width device_width SEC
address [SEC address ...]
```

You must include all of the parameters for this command. The parameters are:

device The erasure algorithm used by the flash device:
Basic for Intel quick-pulse (or the identical AMD Flashrite), erase chip
Intel for Intel Auto, erase selected sectors
12V_AMD for AMD 12V, erase chip
5V_AMD for AMD 5V embedded, erase chip
S_5V_AMD for AMD 5V embedded, erase selected sectors

start_addr The start address in hex. Example: for a hex address of 10000, enter 0x10000, 010000, or 10000.

ROM_size The size of the ROM in hex. Example: for a size of 8000 bytes (hex), enter 0x8000, 08000, or 8000.

bus_width	The width of the bus: **/B** for byte **/W** for word **/L** for long
device_width	The width of the device: **/B** for byte **/W** for word **/L** for long
SEC	When erasing selected sectors, enter "SEC" before each address. A maximum of 4 addresses can be specified.
address	The sector address in hex. Example: to erase sector address 10000, enter 0x10000, 10000, or 010000.

Example

To erase sectors starting at 100000 and 101000 for an AMD 5V Embedded device with a bus width of 4 bytes, a device width of 1 byte, and a ROM size of 32768 bytes, enter:

```
ss ef S_5V_AMD 0x100000 0x8000 /L /B SEC 0x100000 SEC 0x101000
```

3

Configuration

Emulator Configurations

What is a configuration?

For an emulator or HP E3490A Processor Probe to function correctly, it needs some information about your target system. This information might include the type of processor, the processor clock rate, real-time circuitry that can be enabled or disabled, and memory mapping.

What is a configuration file?

When you find a configuration that works for your target system, you can save all of this configuration information in a configuration file. Whenever you start up a debugging session, you should load a configuration file.

HP supplies default configuration files with:

- the emulator/analyzer interface software
- the E3490A option AAV/AAY product for the HP Processor Probe

The default configuration needs to be modified for most target systems.

Configuration files are also supplied with the connection software to work with the demo program. These configuration files can be found in the $HP64000/demo/master directory.

For HP emulators, configuration files customarily use a .EA file name extension.

To load a configuration

Here are several ways to load a configuration:

- Specify a configuration to use when you start the XRAY debugger:

  - as a default argument (page 36)

  - as an argument when starting xhi68khp from the Control Panel (page 35)

  - as a UNIX command line argument (page 37)

- Open the **HP Misc Commands**→**Config...** notebook page. Enter the configuration file name in the text field, or use the file chooser to select a configuration file, then click the **Apply** button.

- From the command entry field in the XRAY debugger Code window, enter:

  `ss config` *filename*

- Load the configuration using the HP emulator/analyzer graphical interface or the Software Performance Analyzer interface.

The easiest way to load a configuration is to set the default arguments for xhi68khp in the Control Panel. Then the configuration will be loaded when you double-click the xhi68khp icon.

When a configuration file has been loaded successfully, the XRAY debugger will display the following message:

`Loaded configuration file:` *filename*

Note
For the HP E3490A Processor Probe, you must configure the target processor type before you use the XRAY debugger.
To configure the processor type, load a configuration file (or modify the configuration immediately after you start the XRAY debugger). Without this configuration, attempts to access processor-specific registers may result in errors.

See also   Page 84 for information on configuration error messages.

To modify a configuration

- In the command entry field in the XRAY debugger Code window, enter:

  ```
  ss config
  ```

  or,

- Open the **HP Misc Commands**→**Config...** notebook page. Click the **Modify Configuration** button, then click the **Apply** button.
- In the HP emulator/analyzer or Software Performance Analyzer interface, select **Modify**→**Emulator Config...**.

For help on any of the configuration items, press F1 while the cursor is over the item.

If are using an HP emulator/analyzer or HP Software Performance Analyzer

If you are using an HP emulator/analyzer or the HP Software Performance Analyzer, refer to the emulator/analyzer graphical interface *User's Guide* for more information on what to configure. Note that the debugger doesn't support display or modification of emulator copies of internal processor registers (EMSIM registers).

If you are using an HP Processor Probe

The next section describes the configuration items which are available for the HP E3490A or HP E3458A Processor Probe.

See also   Page 84 for information on configuration error messages.

Configuring the HP Processor Probe

This section describes the configuration items for the HP E3490A or HP E3458A Processor Probe.

The HP Processor Probe supports only those features that are common on all the processors in the CPU32 family. Additional processor specific features and capabilities are supported through the use of user configurable parameters. These parameters may be customized for specific target systems and saved in configuration files for future use.

What can be configured

The configurable parameters and their default values are:

- The target processor type.

  Default = 683xx

- The target processor clock rate (determines BDM communication speed).

  Default: ≥8 MHz if configuration switch #2 is CLOSED.

  Default: ≥131 kHz if configuration switch #2 is OPEN.

Initial values for memory mapped registers *cannot* be specified as they can with the HP debugger interfaces.

What happens when a configuration is applied

When a configuration is applied to HP E3490A or HP E3458A
Processor Probe, the following events occur:

1 The target is reset.

2 The BDM communication speed is set to the rate based on the
default target processor clock rate.

3 The target processor enters the BDM monitor.

4 The emulation copies of the SIM registers (specified in the
configuration) are loaded into the target SIM.

5 The BDM communication speed is set to the rate based on the value
configured as the target processor clock rate.

6 The list of registers in the XRAY debugger is set to match the target
processor.

To quickly configure the HP Processor Probe

If you have a running target system, and you want to get the HP Processor Probe configured quickly, follow these steps:

1 Connect the HP Processor Probe to your target system.

2 Start the XRAY debugger.

3 Execute the target system's initialization code. In the Code window command entry field, enter:

```
ss reset go
```

This command resets the target processor and begins execution.

4 Click **Stop** to break target execution.

At this point, the target processor's internal registers should have been properly configured by the target system's initialization code.

5 Open the configuration window. In the command entry field in the XRAY debugger Code window, enter:

```
ss config
```

6 In the configuration window, click **General Items**. Enter the target processor you are using and click **OK**.

7 Copy the internal register values from the target system into the HP Processor Probe configuration. In the Emulator Configuration window, select **Modify→Copy SIM to EMSIM to Config**.

The **Reconfigure Internal Registers** configuration section should now be available. Click the button for this section if you wish to examine or change the values of the internal registers.

8 Save the configuration. In the configuration window, select **File→Store…**.

9 To load the configuration automatically when you double-click the **xhi68khp** icon, set this configuration file as a default argument (page 36).

To set the processor type

1 Open the Emulator Configuration dialog box.

2 In the Emulator Configuration dialog box, select **General Items**.

3 Position the mouse pointer in the Processor field.

4 Enter the processor type.

To see a list of processor types supported, press the F1 key.

5 Click on the **OK** button.

N o t e   Once you have set the processor type, you cannot change it unless you exit (**Exit Released**) and restart the XRAY debugger.

If you enter the wrong processor type and have not yet applied the configuration, you can exit the configuration dialog box and start over with step 1 above.

The default value for the processor configuration item is 683xx. With this value, the HP Processor Probe can provide direct access to all the registers defined in the CPU32 architecture programming model but will not have direct access to memory mapped registers in the processor's internal modules.

If a processor is specified, the HP Processor Probe will have knowledge of on-chip peripheral registers and SIM registers and will allow display and modification from the user interface. For example, when 68332 is selected as the processor type, the interface will support direct access to the SIM, the QSM, the TPU, and the TPURAM registers.

The HP Processor Probe does not have explicit support for all CPU32 processors. When using a member of the CPU32 family that is not explicitly supported it may be possible to select a processor that is a formal subset of the unsupported processor. This will provide direct access to all of the internal memory mapped registers that are common. Since the registers in the internal modules are memory mapped, registers in unsupported CPU32 processors are also accessible through the memory commands.

See also   Page 64 for instructions on opening the Emulator Configuration dialog box.

To set the processor clock rate (BDM communication speed)

1 Open the Emulator Configuration dialog box.

2 In the Emulator Configuration dialog box, select **General Items**.

3 Select a Processor Clock Rate.

4 Click on the **OK** button.

The processor clock rate configuration parameter is used to specify the maximum rate that the HP E3490A or HP E3458A Processor Probe can communicate with the target processor through the BDM port. This maximum communication rate is based upon the target processor type and the target processor clock rate. The maximum communication rate directly impacts the performance of downloading files through the HP Processor Probe. Running at a rate slower than the maximum does not prevent correct operation.

The default value for the processor clock rate configuration parameter is determined by the setting of configuration switch S2. In the Normal (CLOSED) position this parameter is set to "greater than or equal to 8 MHz". In the OPEN position this parameter is set to "greater than or equal to 131 kHz".

Most target systems run with a Processor Clock Rate of 8 MHz or above and so will function properly using the default value. For correct operation in target systems that run slower than 8 MHz, the default clock rate must be set to "greater than or equal to 131 kHz" by setting switch S2 to OPEN.

The best performance for the HP E3490A Processor Probe usually requires changing the communication rate from the default value. Set the "Processor Clock Rate" parameter to the highest rate that is equal to or less than the target running clock rate.

See also "Detailed information about processor clock rates" in the *HP E3490A Processor Probe User's Guide*.

To set the HP E3490A default clock rate (< 8 MHz)

• If the processor clock rate is greater than or equal to 8 MHz, set switch S2 to CLOSED.

• If the processor clock rate is less than 8 MHz, set switch S2 to OPEN.

If a target system's processor clock rate is less than 8 MHz following powerup, the default clock rate must be set to "greater than or equal to 131 kHz" by setting configuration switch 2 OPEN.

This can occur when the target system has the processor running off of an external clock source that is less than 8 MHz or is using the clock synthesizer with a crystal that is lower in frequency that the standard crystal.

The actual processor clock rate should then be communicated to the HP Processor Probe through the configuration processor clock rate entry. The HP Processor Probe will then start communications with the target processor at the 131 kHz processor clock rate. When the configuration process is complete, the HP Processor Probe will change the communication rate to a rate based on the "Processor Clock Rate" parameter entry.

The HP E3490A Processor Probe will not communicate correctly with target systems that have a processor clock rate slower than 131 kHz.

Note   The HP Processor Probe does not automatically match the communication speed to the actual target speed (SYNCR register). Maintaining consistency is the responsibility of the user.

To maintain consistency, specify a correct SYNCR register value in the configuration process and make sure that the target code does not change the SYNCR register to a value that is slower than what is specified in the configuration.

Note   The HP Processor Probe will not run correctly if the actual target processor clock rate is slower than the rate specified in the configuration.

See also   "Detailed information about processor clock rates" in the *HP E3490A (HP E3458A) Processor Probe User's Guide*.

4

Interactions Between Interfaces

Which interface to use

The XRAY debugger should be used for *all* tasks except those which are unsupported in the debugger (for example, trace analysis or performance analysis). Thus, the XRAY debugger should be used for run control (stepping, running, stopping, and setting or clearing breakpoints), for loading files, and for displaying source code.

The HP emulator/analyzer graphical interface should be used to analyze bus activity, to set sophisticated triggers to stop execution or capture trace data, and to work with emulator copies of internal registers (EMSIM registers).

The HP Software Performance Analyzer should be used to measure program activity and module duration.

> For best results, use the XRAY debugger whenever possible. Use the other interfaces only when you need to use special emulator capabilities such as trace analysis.

Effects of commands on a second interface

If the XRAY debugger interface and the HP emulator/analyzer graphical interface are running at the same time, issuing a command in one interface may affect the other interface.

Most commands behave as you would expect: changing a directory in one interface changes the directory in the other interface, running the target program in one interface causes the other interface to show it running, and so on.

Sometimes it will take several seconds before the second interface responds.

The tables which follow list commands that behave differently between the XRAY debugger and the HP emulator/analyzer interface.

Effects of XRAY debugger commands on the HP emulator/analyzer interface

If you do this in the XRAY debugger...	the HP emulator/analyzer interface does this...
Load a file	Code and symbols are loaded. Old symbols are cleared and replaced by the new symbols.
Step	Steps
Insert a breakpoint	Breakpoint is visible
Exit	Window is closed

Effects of HP emulator/analyzer commands on the XRAY debugger interface

If you do this in the HP emulator/analyzer interface...	the XRAY debugger does this...
Load a file	Loads symbols only—code window is empty. Breakpoints are restored.
Step	Takes a few seconds to "see" the step and synchronize the display. Do not execute any debugger commands until the display is synchronized.
Insert a breakpoint	No breakpoint is visible, but target code execution will stop when the breakpoint is reached
Exit (window)	No effect
Exit (locked)	Debugger window is closed
Exit (released)	Debugger window is closed

N o t e   If you initiate run control commands (such as stepping or running) in both interfaces, one interface may get out of sync with the other as each interface tries to be in charge of the command execution. For this reason, all run control commands should be entered from the XRAY debugger interface.

Interactions when starting interfaces

If you start the XRAY debugger or the HP emulator/analyzer interface (or HP Software Performance Analyzer) while the other interface is connecting to or configuring the emulator, an error will occur. Wait until the first interface has finished loading the configuration before starting the second interface.

Hint: Start the XRAY debugger first. Wait until the "Loaded configuration file: filename" message has appeared in the Code window output display before starting the HP emulator/analyzer interface.

Interactions when configuring the emulator

Only one configuration process may be active at a time.

Therefore, entering "ss config" (which starts the configuration process) in the XRAY debugger command entry field, followed by "ss config *filename*", will result in an error. The configuration process window must be closed before you can enter "ss config *filename*".

Interactions when setting and clearing breakpoints

Breakpoints set in the XRAY debugger will appear in the HP emulator/analyzer interface, but breakpoints set in the HP emulator/analyzer interface will not appear in the XRAY debugger.

Breakpoints cleared in the XRAY debugger will be cleared in the HP emulator/analyzer interface, but breakpoints cleared in the HP emulator/analyzer interface will still be visible in the XRAY debugger.

If you clear a breakpoint using the HP emulator/analyzer interface, then try to delete the breakpoint from the XRAY debugger, a "Specified breakpoint not in list" error message will be generated; you may safely ignore this message. To avoid seeing this message, always set and delete breakpoints using the XRAY debugger.

Breakpoints will be restored when you load a file. To clear the breakpoints, use the `clear` command in the XRAY debugger.

Interactions when loading files

When you load a file using the HP emulator/analyzer interface, an empty load window will appear briefly in the XRAY debugger interface.

The HP emulator/analyzer interface only retains symbols from the last symbol file loaded.

Effects of register caching

The XRAY debugger caches register values. When registers are changed by executing code, through the XRAY debugger, or through the HP emulator/ analyzer graphical interface, the register values are updated. If, however, a register is changed by some other means (such as through the built-in terminal interface), the cache is not updated and the XRAY debugger will not display the correct value.

The HP emulator/analyzer graphical interface does not cache register values, so it will always display the correct values.

5

Restrictions and Limitations

XRAY Debugger Unsupported Features

The XRAY debugger doesn't support the following features:

Test coverage.   Use the HP B1418 Branch Validator to determine what portions of your code were executed during testing. (Not available for the HP E3490A Processor Probe.)

Performance analysis.   Use the HP B1487 Software Performance Analyzer to measure software performance. (Not available for the HP Processor Probe.)

Trace analysis.   Use the HP emulator/analyzer interface. (Not available for the HP Processor Probe.)

Simulated I/O.   Use the emulator/analyzer interface. (Not available for the HP Processor Probe.)

Register coverage.   Some registers cannot be displayed or modified in the XRAY debugger. These registers can be displayed and modified using the HP emulator/analyzer interface. These registers include:

- 68030 and 68040 floating point registers
- EMSIM registers (emulator copies of SIM registers)
- Communication Processor Module (CPM) registers, such as the RCCR register in the 68360

Online Help   The online help for the XRAY debugger is the same as that for the Microtec Research xhs68k Debugger Simulator. No online help is available for the **setstatus** commands listed on page 44, or for messages generated by the HP software.

XRAY Debugger Unsupported Commands

The commands corresponding to the following notebooks are not supported:

- Utility Commands
- Advanced Commands
- Bus Trace Commands

The following XRAY Debugger commands are not supported:

SAVE	RAMACCESS
RESTORE	ROMACCESS
RIN	TRACE
ROUT	SETSTATUS EVENT
DIN	SETSTATUS READ
DOUT	SETSTATUS WRITE
ICE	SETSTATUS TRACE
NOICE	SETSTATUS TRIGGER
INPORT	SETSTATUS VERIFY
OUTPORT	SETSTATUS QUALIFY
INTERRUPT	STATUS EVENT
NOINTERRUPT	STATUS QUALIFY
BREAKREAD	STATUS BUFFER
BREAKWRITE	STATUS TRACE
PROFILE	STATUS TRIGGER
PRINTPROFILE	
ANALYZE	
PRINTANALYSIS	

See also

The *XRAY Debugger Reference Manual* or the XRAY Debugger online help for a complete list of debugger commands and features.

Other Restrictions and Limitations

- MasterWorks should not be run as a background process.

- All run control commands (such as stepping, running, and breakpoints) should be initiated only from within the XRAY debugger interface.

  The emulator/analyzer interface and the HP Software Performance Analyzer interface should only be used for those commands which are unavailable from the XRAY debugger interface.

- Breakpoints should be cleared rather than disabled from the debugger.

6

Solving Problems

Messages

This section lists some of the messages which may appear when you use the XRAY debugger. This list is not complete. Error messages containing "(HP)" are generated by the HP software or the HP emulator or HP Processor Probe; all other messages are generated by the XRAY debugger.

Start-up Messages

The following messages may appear when you start the XRAY debugger:

Warning: /usr/4lib/libc.so.1.8 has older revision than expected 9.

The SunOS version is older than 4.1.4. This message can be ignored. (This message will be printed to stdout.)

error: binding socket: Invalid argument

There may be a semaphore problem. See page 94. (This message will be printed to stdout.)

ERROR (HP): Communication timed out.

Check that the emulator is turned on. (An emulator which is off or not plugged in can result in communication time-out errors.)

If the emulator is turned on, turn the emulator off, then on again.

Check that the emulator is properly connected to the LAN. To do this, enter:

```
emul700 -lv emulator_name
```

on the UNIX command line. You should see a message which identifies the emulator and its availability.

ERROR (HP): Duplicate connections to the emulator are not allowed!

Exit any other interfaces which are connected to the emulator, then try starting the XRAY debugger.

ERROR (HP): Emulation software not found for this emulator!

You must install the emulator/analyzer graphical interface software before you use the XRAY debugger.

Check that your emulator is listed on page 12.

ERROR (HP): Emulator locked to *user@address*.

The emulator is in use by another user, or another user has left the emulator locked after finishing a session.

Ask the other user to unlock the emulator, or use the emul700 -U command.

ERROR: Exiting the Environment Now! Invalid command line argument [-e emulator_string].

The emulator name required for the **-e** argument is not syntactically correct. Start the XRAY debugger again, specifying the name of an emulator.

You must supply an emulator to connect to using the following syntax:

```
-e emulator_name
```

or

```
-e emulator_name,config_filename
```

Spaces are not allowed between the emulator name and the comma, nor between the comma and the configuration filename.

ERROR (HP): Invalid internet address specified for target.

Check that the IP address for the emulator is correct.

ERROR (HP): Last emulation session was aborted! Initialize emulator before proceeding!

Initialize the emulator by turning it off, then on again.

ERROR (HP): Lost contact with emulator! Initialize emulator before proceeding!

Initialize the emulator by turning it off, then on again. Save any files (configuration files, for example), then exit and restart the XRAY debugger and any other interfaces which may have been connected to the emulator.

ERROR (HP): Old software has been installed! Reinstall latest version of emulation software.

Install a current version of the emulator/analyzer interface software. See page 12 for compatibility information.

ERROR (HP): Shared lib *library_name* **not found**

If the environment variables ($HP64000, $LD_LIBRARY_PATH, $USR_MRI, and $XRAYMASTER) are properly set, and this message still appears, reinstall the product.

If after reinstalling the product, this message still appears, contact HP support.

ld.so: libMWxry.so.4: not found

Check that the LD_LIBRARY_PATH includes the $USR_MRI/master/lib directory.

Configuration Messages

The following messages may appear when you try to load or modify a configuration:

ERROR (HP): A configuration process for the emulator is already running!

Look for another configuration dialog window on your screen (it might be hidden or iconified). You can only run one configuration process at a time.

If you have just started both the XRAY debugger and the HP emulator/analyzer graphical interface (or HP Software Performance Analyzer interface), wait a few seconds for the first interface to finish loading its configuration, then restart the second interface.

Cannot open file: *filename***.EB**

The directory that contains the configuration file you are loading must have write permissions. The .EB file is a temporary file used by the configuration process.

ERROR (HP): Configuration file *filename* **was not found!**

Check that you correctly entered the name of a configuration file. Check that you gave the correct path to the file.

ERROR (HP): Configuration file *filename* **has an invalid status!**

Check that the file exists, that the file has read and write permissions, and that the directory has read and execute permissions.

ERROR (HP): Configuration file *filename* **is not a regular file!**

Check that the file name you specified is the name of a configuration file, and not the name of a directory or another kind of file.

ERROR (HP): Configuration file contains invalid data. File not loaded!

Use another configuration file. For example:

```
ss config anotherconfig.EA
```

If necessary, start over with the default configuration and edit it using the Modify Configuration dialog box:

```
ss config
```

ERROR (HP): Invalid emulator status! Re-cycle power on emulator before continuing.

The HP connection software cannot determine what chip it is connected to. Cycle power on the emulator or HP Processor Probe. If this does not fix the problem, contact HP for assistance.

Register Messages

The following messages may appear when you try to access a register:

ERROR (HP): Register mbar=*value*; valid bit not set
ERROR (HP): Display register failed: @*reg*

Set the MBAR register.

Some registers are mapped to a region of memory specified by the MBAR register. You must set the MBAR register before you can read or write one of these memory-mapped registers.

ERROR (HP): Requested register not supported for this processor

This message will be displayed if you enter the name of a register which is supported for a different processor, but which is not valid for the processor you are using.

Check the list of supported registers beginning on page 48.

Memory Operation Messages

The following messages may appear when you try to access memory:

Copy (Fill, Search, Compare) request stopped by user.

This message means that you clicked the Stop button while the emulator was copying, filling, searching, or comparing a block of memory.

ERROR (HP): Copy memory failed! Specify a range less than 64k bytes.

Enter a range which has a size less than 64k bytes.

ERROR (HP): Copy request failed! Can't allocate space for the copy.

Heap memory could not be allocated for a temporary copy. If you are copying a large range of memory, try copying a smaller range.

ERROR (HP): Fill pattern size exceeds 16k bytes.

Specify a smaller pattern to fill the address space.

ERROR (HP): Search pattern size exceeds 8k bytes.

Specify a pattern smaller than than 8k to search a range of memory.

ERROR (HP): Target memory access failed.

Check the address you are trying to access.

The operation tried to access memory which does not exist, guarded memory, or memory which cannot be accessed for other reasons.

ERROR (HP): Test pattern size exceeds 8k bytes.

Specify a pattern smaller than than 8k to test a range of memory for a mismatch.

NOTE (HP): Long size memory access '/L' is not supported for this processor. Redo this task using a '/W' or '/B' option instead.

Convert the long size access to word or byte and try the operation again.

For some memory operations, long memory accesses are automatically converted to word accesses for the 68000 and 68302 processors. For memory operations which need to return an exact address, this message is displayed.

WARNING (HP): Request access to guarded memory.

Check the address you are trying to access.

The operation tried to access guarded memory. The emulator's memory map specifies which address ranges are guarded. Use the `ss config` command to open the Emulator Configuration dialog box and examine the memory map.

Flash Programming and Flash Erasing Messages

ERROR (HP): Program flash command unsuccessful. Check that you have entered the correct command for your flash part.

Check that the parameters you have specified are correct for your flash part. Re-enter the command with the correct parameters.

ERROR (HP): Flash Programming and Erasing are not available with emulators. Use a Processor Probe to accomplish these tasks!

You are connected to an emulator and this command is only available for Processor Probe connections.

ERROR (HP): Missing sector erase argument(s) in erase flash command for either AMD 5V Embedded or Intel Auto device!

You have selected these two algorithms to use with an erase selected sectors option. You must specify sectors to erase.

ERROR (HP): Invalid start address (or size) argument for program flash (or erase flash)! "0xk" is not a hex value!

Enter a valid hex value for the flash parameter specified.

ERROR (HP): Invalid device argument "bassic" for program flash! Expecting "Basic", "Intel", "5V_AMD", or "12V_AMD".

Re-enter a correct flash device.

ERROR (HP): Specify a Size (bytes) greater than 0x0 for program flash (or erase flash) command!

Must have at least one byte to program or erase! Re-enter a larger value.

ERROR (HP): Invalid bus width (or device width) argument "/m" for program flash (or erase flash)! Expecting /B, /W, or /L.

Valid choices for these two parameters are /B, /W, or /L. Re-enter a correct choice.

Unable to open absolute file: whatever.x.
Terminated when processing absolute file. LOAD /ns 'whatever.x'

These messages are generated by the MRI software if the file you are trying to load int o flash memory does not exist or is not readable.

ERROR (HP): Missing file argument for program flash command!

You must specify a file to load into the flash memory (IEEE695 format only).

ERROR (HP): Too many arguments for erase flash command.
"Erase chip" was selected for this device! Do not specify any
sectors to erase.

You have chosen an algorithm which will erase the chip, yet you have specified sectors to erase. Re-enter the command with the listed sectors removed.

ERROR (HP): Missing arguments for program flash (or erase
flash) command!

Re-enter the command specifying all the required parameters.

ERROR (HP): Syntax error in erase flash command. Missing
parameter SEC!

You have entered the erase flash command on the debugger command line and have forgotten to enter the token "SEC" before the sector address value. Re-enter the erase flash command with the correct syntax.

ERROR (HP): Syntax error in erase flash command. Missing hex
value for sector address!

You have entered the erase flash command on the debugger command line and have forgotten to enter the sector address value after the token "SEC". Re-enter the erase flash command with the correct syntax.

ERROR (HP): Invalid sector address argument for erase flash!
"ox9000" is not a hex value!

You have entered an incorrect hex value for the sector address value. Re-enter the erase flash command with the correct syntax.

ERROR (HP): Too many arguments for erase flash command!

Re-enter the command correctly.

HP Branch Validator Messages

ERROR (HP): Missing arguments for bbaunload command!

Re-enter the command with correct syntax.

ERROR (HP): Missing file argument for bbaunload command!

You need to either select the default file or an alternate file.

**ERROR (HP): Invalid offset value in bbaunload command!
"ox9000" is not a valid hex value.**

You have entered an invalid hex number for the offset value. Re-enter a valid
hex value for this parameter.

ERROR (HP): Missing hex offset value in bbaunload command!

You must enter a hex value for the non-default offset chosen.

**ERROR (HP): Missing file or offset argument in bbaunload
command!**

Not enough parameters specified with the ss bbaunload command. Re-enter
the command using correct syntax.

ERROR (HP): Symbol file not loaded.

No symbol file has beed loaded while using the ss bbaunload command. Load
a symbol file first.

Miscellaneous Messages

ERROR (HP): Could not create semaphore, perhaps kernal limits too low!

See page 94.

ERROR (HP): Specified breakpoint not in list: *address*

Normally, you can ignore this message.

If you delete a breakpoint from the HP emulator/analyzer interface, the breakpoint will not appear to have been deleted in the XRAY debugger. If you then try to delete the breakpoint from the XRAY debugger, this message will be generated.

To avoid seeing this message, always set and delete breakpoints using the XRAY debugger.

WARNING (HP): Unable to break

The emulator was unable to break processor execution. This condition can usually be corrected by resetting the processor (enter "reset" in the command entry field in the Code window).

STATUS (HP): Messages slow or no bus cycles.

This message means that your target system isn't generating any bus activity, usually because it is stopped or because it is running in cache.

Other Sources of Information About Messages

Help on Last Error

Information about error messages generated by XRAY MasterWorks is available with the **Help On Last Error** menu selection.

Information about HP messages is not available with the online help.

In general, HP messages begin with the text "ERROR (HP):", "WARNING (HP):", "STATUS (HP):", or "NOTE (HP):", followed by the text of the message; XRAY MasterWorks messages do not have this format.

Processes

If the XRAY debugger does not terminate properly, it may be necessary to manually kill one or more of these processes.

The following processes will be started by the XRAY debugger:

- **xhi68khp**—the XRAY debugger
- **xti68khp**—interfaces the XRAY debugger with other HP software
- **emul700dmn**—communicates with the emulation hardware
- **xconfig**—the configuration process; only active when loading or modifying a configuration
- **ui-term**—allows communication between the xti68khp process and the emul700dmn process

The following processes may also be started:

- **xui-xemul**—the HP emulator/analyzer graphical interface
- **xui-xperf**—the HP Software Performance Analyzer

Semaphores

If the XRAY debugger is terminated abnormally, semaphores may remain locked. To check for and remove semaphores:

1 Exit the XRAY debugger, the HP interfaces, and any other programs which you suspect may be using semaphores.

2 At the UNIX prompt, enter: `ipcs -s`

This command displays a list of semaphores on your workstation.

3 Find any semaphores which list your login as the owner. Delete them by entering: `ipcrm -s ID`

4 Cycle power on the emulator to reinitialize it.

Example

User "paul" would enter:

```
$ ipcs -s
IPC status from sun2 as of Tue Jan 16 14:05:35 1996
T    ID    KEY         MODE        OWNER    GROUP
Semaphores:
s    545 0x5017b414 --ra-ra-ra-    jerri     100
s   1176 0x50178725 --ra-ra-ra-     dave     100
s   1037 0x7317a21e --ra-ra-ra-     paul     100
s    158 0x50173f15 --ra-ra-ra-   sherry    other
$ ipcrm -s 1037
$ ipcs -s
IPC status from sun2 as of Tue Jan 16 14:07:10 1996
T    ID    KEY         MODE        OWNER    GROUP
Semaphores:
s    545 0x5017b414 --ra-ra-ra-    jerri     100
s   1176 0x50178725 --ra-ra-ra-     dave     100
s    158 0x50173f15 --ra-ra-ra-   sherry    other
$
```

94

Files

HP files

The following files and directories are installed with the HP software:

$HP64000/master/shinsei
$HP64000/master/shinsei/lib/pkbatch.so (SunOS and Solaris systems)
$HP64000/master/shinsei/lib/pkbatch.sl (HP-UX systems)
$HP64000/master/shinsei/messages/default/*.ini
$HP64000/master/shinsei/objects
$HP64000/inst/debug64700/lib/ui-term
    the "ui-term" process
$HP64000/inst/emul/*emulator*
    configuration, emulation, and emulation daemon files
$HP64000/master/mwbin
    the xti68khp process; linked by the $XRAYMASTER/mwbin
directory
$HP64000/master/etc
    bitmaps, support files; linked by the $XRAYMASTER/etc directory
$HP64000/master/xhi68khp
    description files linked by the $USR_MRI/xhi68khp directory
$HP64000/master/dialang
    the hp.dia file used to create the HP notebook pages

The default value for $HP64000 is /usr/hp64000.

Microtec Research files

The following directories are installed with the Microtec Research software:

$USR_MRI/xhi68khp
$XRAYMASTER/bin
$XRAYMASTER/dialang
$XRAYMASTER/etc
$XRAYMASTER/help
$XRAYMASTER/lib
$XRAYMASTER/mwbin

The default value for $USR_MRI is /usr/mri. The default value for
$XRAYMASTER is /usr/mri/master.

Updating firmware

For emulators, instructions for updating firmware can be found in the *Terminal Interface User's Guide* or *Installation Guide*.

For the HP Processor Probe, instructions for updating firmware can be found in the *HP E3490A (HP E3458A) Processor Probe User's Guide*.

Getting Help

Who to call when you have a problem

Whenever possible, try the same command in another interface. For example, if a command doesn't work in the XRAY debugger, try the same command in the XHS68K debugger or in the HP emulator/ analyzer interface. This can help you decide whether the problem is originating with the HP or Microtec Research parts of the software, or with the HP emulator or HP E3490A Processor Probe.

Problems related to HP messages

Call HP if the error message is formatted with the string "(HP)".

Call Microtec Research for other error messages.

Problems related to software installation

If the XHS68K debugger is working, but the XHI68KHP (or XOI68KHP) XRAY debugger is not working, call HP.

If you have problems installing XRAY HP ICE or if the XHS68K debugger does not work, call Microtec Research.

Problems related to emulator connection and configuration

Call HP.

7

About this Version

The following features have been added in this version (07.11) of the HP Connection for Microtec Research MasterWorks XRAY HP ICE.

More Processors

Support for the 68EN302, 68396, 68376, and 68370 processors has been added.

The following features were added in previous versions of the HP Connection for Microtec Research MasterWorks XRAY HP ICE.

Flash Programming

If you are using an HP Processor Probe, you can program and erase flash memory parts in your target system.

HP Branch Validator (BBA) Support

Support has been added for the HP Branch Validator.

HP Notebooks

Interfaces for emulator configuration, flash programming, and branch analysis are now provided through HP notebooks.

These notebooks must be installed by each user.

Support for 64700tab.net

Aliases for emulator or HP Processor Probe LAN addresses can be defined in the 64700tab.net file. The connection software now supports these aliases.

For information on how to use the 64700tab.net file, refer to the *Graphical User Interface User's Guide* for your emulator.

Index

Document Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety Symbols

WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.